Distance sets for shape filters and shape recognition

Grigorescu, Cosmin; Petkov, Nicolai

Published in:
Ieee transactions on image processing

DOI:
10.1109/TIP.2003.816010

Distance Sets for Shape Filters and Shape Recognition

Cosmin Grigorescu, *Student Member, IEEE,* and Nicolai Petkov

***Abstract*—We introduce a novel rich local descriptor of an image point, we call the (labeled) distance set, which is determined by the spatial arrangement of image features around that point. We describe a two-dimensional (2-D) visual object by the set of (labeled) distance sets associated with the feature points of that object. Based on a dissimilarity measure between (labeled) distance sets and a dissimilarity measure between sets of (labeled) distance sets, we address two problems that are often encountered in object recognition: object segmentation, for which we formulate a distance sets shape filter, and shape matching. The use of the shape filter is illustrated on printed and handwritten character recognition and detection of traffic signs in complex scenes. The shape comparison procedure is illustrated on handwritten character classification, COIL-20 database object recognition and MPEG-7 silhouette database retrieval.**

***Index Terms*—Character recognition, distance set, image database retrieval, MPEG-7, object recognition, segmentation, shape descriptor, shape filter, traffic sign recognition.**

I. Introduction

THE world, as we visually perceive it, is full of information that we effortlessly interpret as colors, lines, edges, contours, textures, etc. It is not only the mere presence of these bits and pieces of information that plays a role in our perception, but also their spatial interrelations which enable us to distinguish between surrounding objects. Let us consider, for example, the binary images shown in Fig. 1(a). The black pixels taken individually, especially those in the interior of the objects shown, do not carry any information whether they are part of a representation of a bird or a rabbit; it is their spatial interrelations which make us recognize the shape of a bird or a rabbit. When only the contours or even parts of the contours of the objects are left, like in Fig. 1(b) and (c), we are still able to discriminate between the two objects. In this particular case, it is the edge points belonging to the contours which are perceptually important and it is their spatial interrelations which define our perception of two different objects. A failure of the structures of the brain responsible for the detection of elementary features, such as lines and edges, or of the higher structures responsible for setting relations between these features and integrating them into perception of specific objects leads to a medical condition referred to by neurologists as form blindness or object agnosia [1]. While many aspects of detection of elementary visual features by the brain

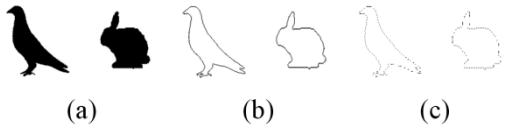

Fig. 1. (a) Two familiar objects, (b) their contours, and (c) parts of their contours.

are fairly well understood [2]–[11], and a large number of algorithms for feature extraction have been developed in computer vision (see, e.g., [12]–[16]), the neurophysiology of feature integration for object perception and recognition is still unknown and the development of computer vision algorithms for this purpose is an ongoing activity [17]–[21].

In this paper, we propose to describe the spatial interrelations between perceptually significant points, to be called feature points, by associating with each such point a data structure containing the set of distances to a certain number of neighboring feature points. As features can be of different types, we associate different labels with such distances and call the resulting data structure the *labeled distance set*. We propose the set of distance sets associated with all feature points of an object as a data structure integrating information about the spatial interrelations of elementary features and show that this data structure can effectively be used to segment objects in complex scenes and to classify objects.

A number which characterizes the image content in the surroundings of a point is often called a *local image descriptor*. The value of a pixel in a grey-level image can be considered as the simplest type of local descriptor, comprising merely one number. An elaborate local descriptor can indicate the presence of a perceptually important feature, often called a salient feature, such as an edge [22], a corner, or a specific type of junction [23] at a given location in the image.

A *rich descriptor* is a set of numbers or, more generally, a data structure computed on the neighborhood of a point and presents a more informative characteristic of the local image contents. The adjective "rich" refers to the use of many values as opposed to one single value, with the degree of richness being related to the size (cardinality) of the descriptor and its complexity as a data structure. Kruizinga and Petkov [24] used as a rich local descriptor a set of arrays of intensity values visible in multiscale windows centered on the concerned point. Wiskott *et al.* [25] considered as a local descriptor a vector of complex Gabor wavelet coefficients computed for several scales and orientations. Belongie *et al.* [20] proposed as a local descriptor of a point the two-dimensional (2-D) histogram (with arguments log-distance and polar angle in relative image coordinates) of the number of object contour points in the surroundings of the concerned point. Amit *et al.* [26] used a set of tags,

Manuscript received August 14, 2002; revised March 17, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Robert D. Nowak.

The authors are with the Institute of Mathematics and Computing Science, University of Groningen, 9700 AV Groningen, The Netherlands (e-mail: cosmin@cs.rug.nl; petkov@cs.rug.nl).

Digital Object Identifier 10.1109/TIP.2003.816010

each tag specifying the occurrence of a given combination of binary pixel values at certain positions relative to the concerned point. The *(labeled) distance set* we propose below comprises multiple values and thus also falls in the class of rich local image descriptors.

Local descriptors are used for different purposes; one can, for instance, select only those points of an image whose local descriptors fulfill a certain condition. For grey-level images, thresholding can be considered as such an operation; the condition to be fulfilled by the local descriptor (in this case the grey-level value of a pixel) is that it is larger than a given threshold value. This type of use of local descriptors can be regarded as filtering: a result image that represents a binary map of points which satisfy a given condition versus points that do not fulfill the condition is computed from an input image.

In the following we define a dissimilarity measure between (labeled) distance sets and, based on this measure, we introduce a novel filtering procedure aimed at detecting instances of a given object (or parts of an object) in a complex scene. We refer to the proposed filtering method as *distance set shape filtering*.

Another use of local descriptors, typical of *rich* local descriptors, is for solving the correspondence problem, i.e., finding counterpart points in two images. We address this problem by representing an object as a set of (labeled) distance sets and by computing a dissimilarity measure between two such sets. We use this measure for pairwise shape comparison and classification.

Image descriptors based on distances between image features have been previously proposed in the literature for object detection [22], [27], shape comparison [20], [28], handwritten digit classification [26], face recognition [29]. Despite this, the concept of a (labeled) distance set, and the associated filtering operator and shape comparison method are, to our best knowledge, novel.

A brief overview of other methods for shape extraction and shape comparison methods based on rich local descriptors is given in Section II. In Section III we introduce the concepts of a *distance set* and a *labeled distance set*, and associated dissimilarity measures. Sets of (labeled) distance sets together with a measure of dissimilarity between two sets of (labeled) distance sets are introduced in Section IV. The distance sets shape filter is introduced in Section V; this filter is related to certain morphological filters. We illustrate the applicability of the distance sets shape filtering to handwritten character recognition and the detection of instances of a given object in complex scenes.

A shape comparison procedure based on a dissimilarity measure between two sets of (labeled) distance sets is presented in Section VI. We evaluate the performance of the proposed comparison method in three applications: handwritten character classification, COIL-20 database object recognition, and MPEG-7 silhouette database retrieval. Section VII summarizes the results and concludes the paper.

II. Overview of Other Shape Extraction and Shape Comparison Methods

We consider shape as a property or characteristic of a set of points that define a visual object whereby said property is determined by the geometric relations between the involved points in such a way that it is invariant for translations, rotations, reflections and distance scaling of the point set as a whole.

A substantial body of work in shape analysis assumes that shape is a characteristic of a binary image region. Such methods use either the boundary of an object or its interior to compute a shape descriptor. Their applicability is envisaged mainly in situations in which a binary object is already available or can be computed by some preprocessing steps like pixel-based segmentation, edge detection, skeletonization, etc. An overview of such shape analysis methods can be found in [17] and [30].

Recent developments in shape analysis describe shape more generally, as a property of a collection of feature points, and associate with each such point a local image descriptor. These descriptors are subsequently used to find the occurrences of a reference object in an image (segmentation), or to evaluate how similar two objects are (comparison). When comparing two objects, one often tries to determine a transformation which casts one of them into another. Since rigid, affine or projective transformations are sensitive to irregular shape deformations or partial occlusion, a more general model, that of nonrigid transformations, is assumed. With this formulation, shape comparison implies either finding pairs of corresponding feature points (i.e., solving a correspondence problem) and/or determining the transformation which maps one point set into the other. A shape (dis)similarity measure can be computed from the solution of the correspondence problem and/or from the nonrigid transformation.

As the proposed (labeled) distance set shape descriptor, the associated shape filtering operator and the shape comparison procedure we propose are based on sets of feature points, we restrict our overview only to similar methods—*nonrigid point-based shape analysis methods*—previously reported in the literature.

Non-rigid shape matching methods have been introduced in computer vision by Fischler and Elschlager [31], who formulated the shape matching as an energy minimization problem in a mass-spring model. A feature-based correspondence approach using eigenvectors is presented by Shapiro and Brady in [32]. Modal matching proposed by Sclaroff and Pentland [33] describes shape in terms of generalized symmetries defined by object's eigenmodes. The shape similarity between two objects is expressed by the amount of modal energy deformation needed to align the objects. Chui and Rangarajan [34] use an integrated approach for solving the point correspondence problem and finding a nonrigid transformation between points from the contours of two objects. Their iterative regularization procedure uses a softassign [35], [36] for the correspondences and a thin-plate spline model [37] for the nonrigid mapping. The method proves to be robust to noise and outliers.

Other approaches select only a number of points from the outline of an object and approximate it with curve segments which pass through those points. Usually, these points have some specific properties, such as minimum curvature, inflections between convex and concave parts of an object contour, etc. Petrakis *et al.* [38], for instance, extract only inflection points and approximate the shape by B-splines curve segments. A dissimilarity cost of associating a group of curve segments from a source shape to another group of curve segments originating from a target shape is computed by dynamic programming. In a similar way, Sebastian *et al.*

[21] use length and curvature to define a similarity metric between curve segments. This metric is subsequently employed by a dynamic programming algorithm to solve the optimum alignment (correspondence) problem. Such methods have a limited applicability because they can be used for the pairwise comparison of single curves only. Objects that are defined by multiple curves, e.g., an icon of a face with separate curves for head outline, nose, mouth and eyes, cannot be compared since the mutual geometric arrangement of the constituent curves that define such an object is not taken into account.

Shape characterization based on distances between points which do not necessarily originate from the object outline is used by a number of authors. A shape extraction method based on the Hausdorff distance between two point sets is proposed by Huttenlocher *et al.* [22], [27]. Although this method does not explicitly find the point correspondences, it is capable of detecting multiple occurrences of the same reference object even in case of partial matching caused by occlusion and clutter. The discrimination is improved by introducing as additional information the local edge orientation [39]. Amit and Kong [40] find geometric arrangements of landmark points in an image by matching decomposable graphs. An extended approach [26], [41] considers as local descriptors topographic codes and compares the descriptors originating from different shapes using multiple classification trees. Gavrila [42] proposes a method for shape extraction based on the distance transform of the edge map of an image. The correlation between this distance transform and the edge map of a reference object is an indicator whether the reference object is present or not in the image. Since the distance transform assigns to each image point as a local descriptor the distance to the nearest edge pixel, this formulation can be considered as a special case of a distance set descriptor with cardinality one.

A local descriptor based on the spatial arrangement of feature points in a fixed-size neighborhood of a point is the shape context of Belongie *et al.* [20], [28]. The shape context of a point is a two-dimensional histogram (with axes the log-distance and polar angle) of the number of contour points in the surroundings of the concerned point. These authors developed a method for shape comparison by first finding the point correspondences as the solution of an assignment problem and then evaluating a nonlinear alignment transformation using a thin-plate spline model. With respect to radial (distance) information, the shape context may be regarded as a simplified, coarser version of the distance set descriptor: the coarseness is due to the histogram binning process. As to the angular information present in the shape context, such information can be included in a labeled distance set by assigning to each distance an orientation label—the polar angle of the concerned feature point. As proposed in [20], the shape context uses only contour information and is intended only for shape comparison. In contrast, the labeled distance set descriptor incorporates multiple feature types. Next to shape comparison, we use this descriptor also for shape segmentation.

III. Distance Sets and Labeled Distance Sets

A. Distance Sets

Let $S = \{p_1, p_2, \ldots, p_n\}$ be a set of perceptually significant points in an image, which we will call *feature points*. The spatial relation of a given feature point to other feature points can be characterized by the set of its distances to these other points. For a given point $p \in S$ one can select only a subset of $N \leq n$ nearest neighboring points and still have a good description of its spatial relation to the other points[1]. Let $d_i(p) \in \mathbb{R}$ be the distance between point p and its i-nearest neighbor from S, $1 \leq i \leq N$. We call the local descriptor

$$\mathcal{DS}_{S,N}(p) = \{d_1(p), d_2(p), \ldots, d_N(p)\}, \tag{1}$$

the *distance set*, more precisely the N-distance set, of point p to its first N nearest neighbors within S. Note that the distance set of a point is not affected by a rotation of the image as a whole.

Given two points $p \in S_1$ and $q \in S_2$ from two images and their associated distance sets $\mathcal{DS}_{S_1,N_1}(p)$ and $\mathcal{DS}_{S_2,N_2}(q)$, $N_1 \leq N_2$, we define the relative difference between the i-neighbor and j-neighbor distances of p and q, respectively, as[2]

$$\epsilon_{i,j}(p,q) = \frac{|d_i(p) - d_j(q)|}{\max(d_i(p), d_j(q))}, \quad 1 \leq i \leq N_1, \ 1 \leq j \leq N_2. \tag{2}$$

Let $\pi(i)$ be a one-to-one mapping from the set $\{1, 2, \ldots, N_1\}$ to the set $\{1, 2, \ldots, N_2\}$, such that $\pi(i_1) \neq \pi(i_2), \forall\, i_1 \neq i_2$, and let Π be the set of all such mappings.

We introduce the *dissimilarity between two distance sets* $\mathcal{DS}_{S_1,N_1}(p)$ and $\mathcal{DS}_{S_2,N_2}(q)$ as follows:

$$\mathcal{D}_{S_1,N_1;S_2,N_2}(p,q) = \min\left\{ \frac{1}{N_1} \sum_{i=1}^{N_1} \epsilon_{i,\pi(i)}(p,q) \mid \pi \in \Pi \right\}. \tag{3}$$

The dissimilarity is thus the cost of the optimal mapping of the distance set $\mathcal{DS}_{S_1,N_1}(p)$ onto the distance set $\mathcal{DS}_{S_2,N_2}(q)$.

Iff $\mathcal{D}_{S_1,N_1;S_2,N_2}(p,q) = 0$, the distance set $\mathcal{DS}_{S_1,N_1}(p)$ is a subset of $\mathcal{DS}_{S_2,N_2}(q)$:

$$\mathcal{D}_{S_1,N_1;S_2,N_2}(p,q) = 0 \Leftrightarrow \mathcal{DS}_{S_1,N_1}(p) \subseteq \mathcal{DS}_{S_2,N_2}(q). \tag{4}$$

In the special case $N_1 = N_2 = N$, the two sets are identical:

$$\mathcal{D}_{S_1,N;S_2,N}(p,q) = 0 \Leftrightarrow \mathcal{DS}_{S_1,N}(p) \equiv \mathcal{DS}_{S_2,N}(q). \tag{5}$$

In this way, the quantity $\mathcal{D}_{S_1,N_1;S_2,N_2}(p,q)$ indicates how dissimilar two points p and q are with respect to their distances to feature points in their respective neighborhoods.

The distance set $\mathcal{DS}_{S,N}(p)$ of a point p, together with the dissimilarity measure defined above, can be an effective means for discrimination of points according to their similarity to a given point. As an example, we consider as feature points the pixels which define printed characters on a computer screen. Fig. 2(a) shows a set of points S_1 that define the character "a," a point $p \in S_1$ from this set and its associated distance set $\mathcal{DS}_{S_1,N_1}(p)$ to the first $N_1 = 5$ neighbors. (Euclidian distance is used in this and the following examples unless otherwise specified.) Fig. 2(b) shows points from the word "alphabet," that comprise a set S_2. For each point $q \in S_2$ we compute its corresponding distance set $\mathcal{DS}_{S_2,N_2}(q)$. In Fig. 2(b), the points for

[1]The point p need actually not be a point from S, it can be any point in the image.

[2]The use of relative differences is motivated by Weber-Fechner law of perception. Alternative definitions with similar effect are possible, e.g., $|\ln\, d_i(p) - \ln\, d_j(q)|$, but we consider this as a technical detail.

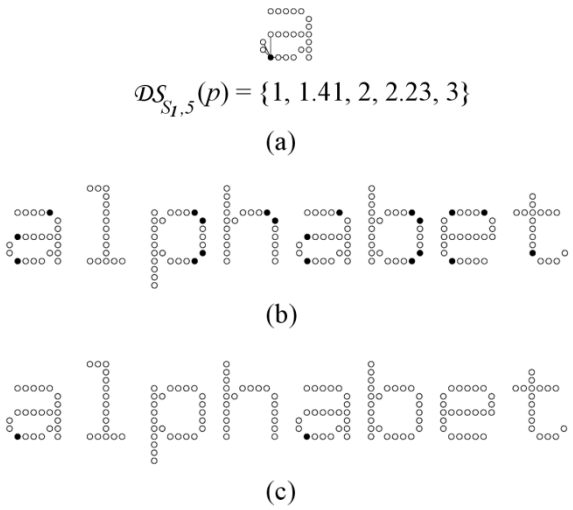

Fig. 2. (a) A point $p \in S_1$ (shed black) from the printed character "a" together with its set $\mathcal{DS}_{S_1,N_1}(p)$ of distances to the first $N_1 = 5$ nearest neighbors within S_1. (b) The points $q \in S_2$ (shed also black) in the word "alphabet" which have 5-distance sets (within S_2) identical with that of p, $\mathcal{D}_{S_1,5;S_2,5}(p,q) = 0$. (c) Points $q \in S_2$ for which $\mathcal{D}_{S_1,8;S_2,8}(p,q) = 0$, the same points are obtained also for $\mathcal{D}_{S_1,15;S_2,20}(p,q) = 0$.

which $\mathcal{D}_{S_1,5;S_2,5}(p,q) = 0$ holds are shown black. From the total of 173 points of the word "alphabet," only 20 have 5-distance sets identical to that of the concerned point of the character "a." The discrimination is improved by increasing the size of the involved distance sets, Fig. 2(c): if, for instance, the distance sets contain $N_1 = N_2 = 8$ distances, only the real counterparts of the selected point are found to have the same distance sets.

In a second example, we consider a point $p \in S_1$, where S_1 is the set of points comprising the contour of a handwritten character "a," and the distance set $\mathcal{DS}_{S_1,15}(p)$ associated with it, Fig. 3(a). The set S_2 consists of the points from the contour of a handwritten word "alphabet." For each $q \in S_2$ we compute the associated distance set $\mathcal{DS}_{S_2,20}(q)$ and its dissimilarity $\mathcal{D}_{S_1,15;S_2,20}(p,q)$ to the distance set $\mathcal{DS}_{S_1,15}(p)$ of the concerned point p from the handwritten character "a." The requirement $\mathcal{D}_{S_1,15;S_2,20}(p,q) = 0$, which was used in the printed character example given above, turns out to be too strong and not fulfilled by any point $q \in S_2$. A weaker requirement $\mathcal{D}_{S_1,15;S_2,20}(p,q) < \theta$, $\theta > 0$, can be imposed on the points of S_2. For $\theta = 0.3$, only 35 from the total number of 216 points of the word "alphabet" fulfill this requirement; the corresponding points are shed black in Fig. 3(b). If an even stricter condition is imposed, $\theta = 0.25$, only 11 points are found to satisfy the condition, Fig. 3(c).

B. Labeled Distance Sets

The feature points in an image can be of different types. In the character "c," for instance, one can consider the points at the extremities of the contour to be of an "end-of-line" type and all other points to be of another, "line" or "contour" type. As a matter of fact, similar, evidently perceptually significant features are extracted in the visual cortex by so-called *simple* and *complex cells* (for lines and edges) [2], [4]–[6], and *end-stopped cells* (for ends of lines) [3], [4], [7]–[9]. The distance

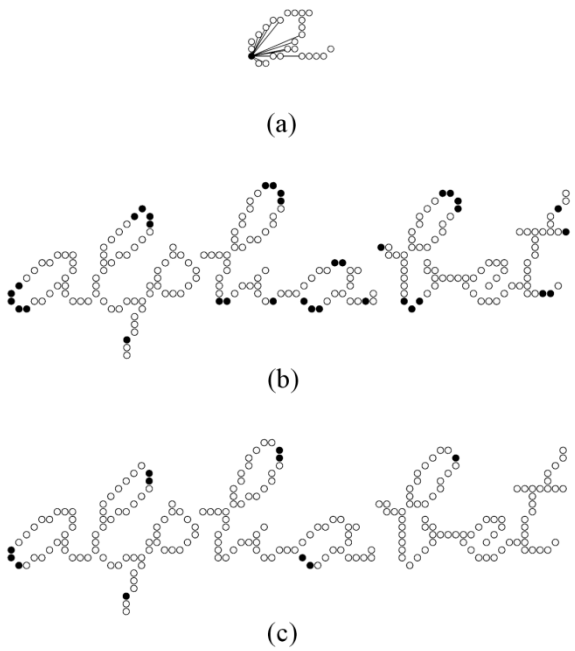

Fig. 3. (a) A point $p \in S_1$ (shed black) from the handwritten character "a" together with its distance set $\mathcal{DS}_{S_1,N_1}(p)$ to the first $N_1 = 15$ nearest neighbors within S_1. (b-c) The points $q \in S_2$ in the handwritten word "alphabet" (shed also black) for which holds (b) $\mathcal{D}_{S_1,15;S_2,20}(p,q) < 0.3$ and (c) $\mathcal{D}_{S_1,15;S_2,20}(p,q) < 0.25$.

from a feature point to another feature point can thus be labeled by the type of feature to which the distance is measured.

Let $\mathcal{L}$ be the set of possible feature labels and let $l \in \mathcal{L}$ be one such label. We define the *labeled distance subset* $\mathcal{DS}^{(l)}_{S^{(l)},N^{(l)}}(p)$ of a point p to its first $N^{(l)}$ neighbor feature points of type l as follows:

$$\mathcal{DS}^{(l)}_{S^{(l)},N^{(l)}}(p) = \left\{ d^{(l)}_1(p), d^{(l)}_2(p), \ldots, d^{(l)}_{N^{(l)}}(p) \right\} \quad (6)$$

where $S^{(l)}$ is the set of feature points of type l and $d^{(l)}_i(p)$ is the distance from point p to its i-th nearest neighbor feature point from that set.

A *labeled distance set* is the set of tuples of labeled distance subsets and their corresponding labels:

$$\mathcal{LDS}_{S^{(l_1)},N^{(l_1)};S^{(l_2)},N^{(l_2)};\ldots;S^{(l_k)},N^{(l_k)}}(p) = \left\{ (\mathcal{DS}^{(l)}_{S^{(l)},N^{(l)}}(p), l) \mid l \in \mathcal{L} \right\} \quad (k = \mathrm{card}(\mathcal{L})). \quad (7)$$

Let $\mathcal{D}^{(l)}_{S^{(l)}_1,N^{(l)}_1;S^{(l)}_2,N^{(l)}_2}(p,q)$ be the dissimilarity between two labeled distance subsets of type $l \in \mathcal{L}$ computed according to (3). Since (in a given context) certain feature points can be perceptually more important than others, for each label type $l \in \mathcal{L}$ we can assign a different weight $w_l \in \mathbb{R}$ and define the *dissimilarity between two labeled distance sets* $\mathcal{LDS}_{S^{(l_1)}_1,N^{(l_1)}_1;\ldots;S^{(l_k)}_1,N^{(l_k)}_1}(p)$ and $\mathcal{LDS}_{S^{(l_1)}_2,N^{(l_1)}_2;\ldots;S^{(l_k)}_2,N^{(l_k)}_2}(q)$ associated with p and q as follows:

$$\mathcal{LD}_{S^{(l_1)}_1,N^{(l_1)}_1;\ldots;S^{(l_k)}_1,N^{(l_k)}_1;S^{(l_1)}_2,N^{(l_1)}_2;\ldots;S^{(l_k)}_2,N^{(l_k)}_2}(p,q) = \frac{\sum_{l \in \mathcal{L}} w_l \mathcal{D}^{(l)}_{S^{(l)}_1,N^{(l)}_1;S^{(l)}_2,N^{(l)}_2}(p,q)}{\sum_{l \in \mathcal{L}} w_l}. \quad (8)$$

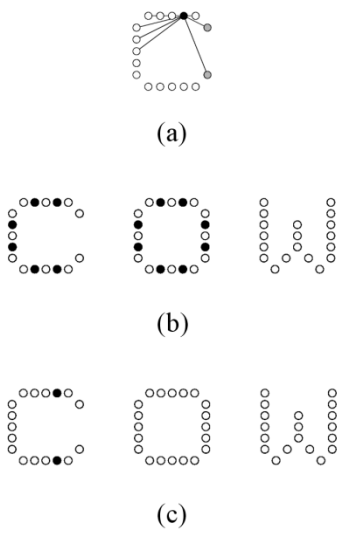

Fig. 4. (a) A point p (shed black) from the letter "c" and its labeled distance subsets $\mathcal{DS}^{(l_1)}_{S_1(l_1),N(l_1)}(p)$ to the first $N^{(l_1)}=7$ nearest neighbors of *contour* type (white) and $\mathcal{DS}^{(l_2)}_{S_1(l_2),N(l_2)}(p)$ to the first $N^{(l_2)}=2$ nearest neighbors of *end-of-line* type (grey). (b) Identical points (regarding their 9-distance sets) from the word "cow" when no feature type labels are taken into account. (c) Identical points when labeled distances are used.

As an example, let us consider a point p from the set of points S_1 defining the printed character "c" on a computer screen, Fig. 4(a), and the set of points S_2 defining the word "cow," Fig. 4(b). We assign two types of labels: l_1—*a regular point*, associated with a pixel coming from the contours of the letters, and l_2—*an end-of-line point* at the extremities of the contours. We build both the distance set (for $N=9$) and the labeled distance set (for $N^{(l_1)}=7$, $N^{(l_2)}=2$) of the point p and for each point of the word "cow." Fig. 4(a) shows the selected point p together with its labeled distance subsets $\mathcal{DS}^{(l_1)}_{S^{(l_1)},N^{(l_1)}}(p)$ and $\mathcal{DS}^{(l_2)}_{S^{(l_2)},N^{(l_2)}}(p)$; the unlabeled distance set $\mathcal{DS}_{S_1,N}(p)$, $N=N^{(l_1)}+N^{(l_2)}$, is the union of these two subsets. Fig. 4(b) presents points from the word "cow" that are identical with the concerned point p regarding their unlabeled 9-distance sets; several such points are found in the character "o." If, however, *labeled* distance sets are used with $N^{(l_1)}=7$ and $N^{(l_2)}=2$, the number of similar points decreases to only two, the real counterparts of the selected point p, as shown in Fig. 4(c). This example illustrates that labeled distance sets can be effective means of improving the discrimination possibilities offered by distance sets.

C. Implementation

The N-distance set of a point p can be determined by selecting the N smallest distances from the n distances of p to all n feature points of the corresponding set S. The computational complexity of this straightforward approach is $O(n)$, and depends on the number of points of the set S. However, by preprocessing and usage of appropriate data structures, the computational complexity of this search problem can be reduced and made dependent only on the number N of neighbors of p that are involved in an N-distance set.

Advanced data structures which can be used to efficiently search nearest neighbors in higher dimensional spaces include quad-trees [43], *K-d* trees [44], range trees [45], lookup maps [46], etc. In order to reduce the computational complexity, some of these data structures make use of an estimate of an underlying probability distribution function which characterizes the occurrence of the points in the search space.

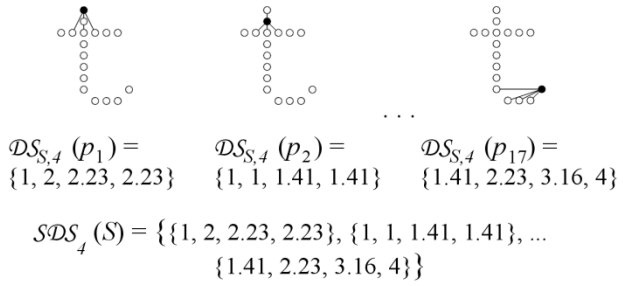

Fig. 5. Set of 4-distance sets $\mathcal{SDS}_4$ of a printed character "t."

The case of distance sets computation is a particular case of a nearest neighbor search in two dimensions. An efficient data structure for finding the first nearest neighbor in two dimensions in amortized constant time $\Theta(1)$, is the *semidynamic binary search tree* (*SD* tree) [47], which is a particular case of a *K-d* tree. The term "semidynamic" refers to the class of trees for which update operations such as deletion and undeletion of nodes are allowed, but insertion of new nodes is not allowed [44]. The computational overhead of building the *SD* tree is $O(n\log(n))$, n being the total number of points. Using this approach, the first N nearest neighbors of a point p and the set of distances to these points $DS_{S,N}(p)$ can be found in constant amortized time $\Theta(N)$.

IV. Sets of (Labeled) Distance Sets

A. Definition

In Section III we suggested that the (labeled) distance set of a point $p\in S$, and the measure of dissimilarity between (labeled) distance sets are suitable means for discrimination. However, a (labeled) distance set describes a local arrangement of features around one point, and not the global spatial arrangement of all the points from the set S. For this latter purpose we now introduce a new concept.

Given the distance sets $\mathcal{DS}_{S,N}(p), \forall\, p\in S$, we define the *set of distance sets* of S as

$$\mathcal{SDS}_N(S)=\{\mathcal{DS}_{S,N}(p)\mid p\in S\} \tag{9}$$

Fig. 5 illustrates this concept.

Similarly, the *set of labeled distance sets* is defined by

$$\begin{aligned}\mathcal{SLDS}_{S^{(l_1)},N^{(l_1)};\ldots;S^{(l_k)},N^{(l_k)}}(S)&\\ =\left\{\mathcal{LDS}_{S^{(l_1)},N^{(l_1)};\ldots;S^{(l_k)},N^{(l_k)}}(p)\mid p\in S, S\right.&\\ \left.=\bigcup_{l\in\mathcal{L}}S^{(l)}\right\}.&\end{aligned} \tag{10}$$

B. Dissimilarity Between Sets of (Labeled) Distance Sets

Given two sets of points S_1 and S_2, $|S_1|\le|S_2|$, we introduce in the following a dissimilarity measure between their associated sets of N-distance sets $\mathcal{SDS}_N(S_1)$ and $\mathcal{SDS}_N(S_2)$.

Let $M: S_1 \mapsto S_2$ be a one-to-one mapping from S_1 to S_2, $|S_1| \leq |S_2|$ ($\forall\, p, p' \in S_1, p \neq p', M(p) \neq M(p')$), and let $\mathcal{M}$ be the set of all such mappings. Let C be a positive constant. We define the *cost of a mapping* $M \in \mathcal{M}$ as follows:

$$\mathcal{C}_N^{(M)}(S_1, S_2) = \frac{1}{|S_2|}\left(\sum_{p \in S_1} \mathcal{D}_{S_1,N;S_2,N}(p, M(p)) + (|S_2| - |S_1|)C\right). \quad (11)$$

The term $(|S_2| - |S_1|)\,C$ is a penalty for those points of S_2 which are not images of points of S_1. Finally, we define the *dissimilarity between the sets of distance sets* S_1 and S_2 as follows:

$$\mathcal{E}_N(S_1, S_2) = \min\left\{\mathcal{C}_N^{(M)}(S_1, S_2) \mid M \in \mathcal{M}\right\}. \quad (12)$$

It holds:

$$\begin{aligned}
&\mathcal{E}_N(S_1, S_2) = 0 \Leftrightarrow \\
&\quad |S_1| = |S_2| \wedge \exists M \in \mathcal{M}\colon \mathcal{C}_N^{(M)}(S_1, S_2) = 0 \Leftrightarrow \\
&\quad \forall\, p \in S_1, \exists! M(p) \in S_2\colon \mathcal{D}_{S_1,N;S_2,N}(p, M(p)) = 0 \Leftrightarrow \\
&\quad \forall\, p \in S_1, \exists! M(p) \in S_2\colon \mathcal{DS}_{S_1,N}(p) \equiv \mathcal{DS}_{S_1,N}(M(p)) \Leftrightarrow \\
&\quad \mathcal{SDS}_N(S_1) \equiv \mathcal{SDS}_N(S_2). \quad (13)
\end{aligned}$$

Similarly, in the case of *sets of labeled distance sets*, $\mathcal{SLDS}_{S_1^{(l_1)},N^{(l_1)};\ldots;S_1^{(l_k)},N^{(l_k)}}(S_1)$ and $\mathcal{SLDS}_{S_2^{(l_1)},N^{(l_1)};\ldots;S_2^{(l_k)},N^{(l_k)}}(S_2)$, the cost of a mapping $M \in \mathcal{M}$, $M: S_1 \mapsto S_2$ is defined as

$$\begin{aligned}
&\mathcal{LC}^{(M)}_{N^{(l_1)}\ldots N^{(l_k)}}(S_1, S_2) \\
&\quad = \frac{1}{|S_2|}\left(\sum_{p \in S_1} \mathcal{LD}_{S_1{}^{(l_1)},\ldots,N^{(l_k)};S_2{}^{(l_1)},\ldots,N^{(l_k)}}(p, M(p)) + (|S_2| - |S_1|)C\right). \quad (14)
\end{aligned}$$

The *dissimilarity between two labeled sets of distance sets* is defined as follows:

$$\begin{aligned}
&\mathcal{LE}_{N^{(l_1)}\ldots N^{(l_k)}}(S_1, S_2) \\
&\qquad = \min\left\{\mathcal{LC}^{(M)}_{N^{(l_1)}\ldots N^{(l_k)}}(S_1, S_2) \mid M \in \mathcal{M}\right\}. \quad (15)
\end{aligned}$$

Similarly to (13), it holds

$$\begin{aligned}
&\mathcal{LE}_{N^{(l_1)}\ldots N^{(l_k)}}(S_1, S_2) = 0 \Leftrightarrow \\
&\mathcal{SLDS}_{S_1{}^{(l_1)},N^{(l_k)}\ldots S_1^{(l_k)},N^{(l_k)}}(S_1) \\
&\quad \equiv \mathcal{SLDS}_{S_2{}^{(l_1)},N^{(l_1)}\ldots S_2{}^{(l_k)},N^{(l_k)}}(S_2). \quad (16)
\end{aligned}$$

The problem of finding the correspondences between the points of two discrete sets S_1 and S_2 is thus reduced to finding an optimal mapping which minimizes the measures according to (11) or (14). In Section VI we show how the cost of the optimal mapping can be used for shape comparison and classification.

C. Implementation

The computation of the *dissimilarity between two distance sets*, as well as the computation of the *dissimilarity between sets of distance sets* involve finding the solution of the optimization problems introduced by (3) and (12). These problems are instances of the square assignment problem, whose optimal solution can be found using the Hungarian method [48] or variants of it [49]. However, both problems can be reformulated in terms of minimum weight assignment problems in a bipartite graph and solved efficiently in $O(v * (e + v\log(v)))$ [47] (where v and e are the number of vertices and edges of the associated graphs, respectively). For instance, in the case of computation of the dissimilarity between two distance sets introduced by (3), the distances $d_i(p) \in \mathcal{DS}_{S_1,N_1}(p)$ are considered as source nodes and $d_j(q) \in \mathcal{DS}_{S_2,N_2}(q)$ are considered as target nodes in an associated bipartite graph. The threshold condition imposed on the dissimilarity between distance sets (Fig. 3), as well as the fact that, within a distance set, the distances are ordered increasingly, leads to a smaller number of correspondences (edges) in the associated bipartite graph and, consequently, to smaller computational complexity. In our experiments, the worst case computational complexity is in the order of $O({N_1}^2\log(N_1))$. A further reduction of the computational complexity of the optimization problem according to (3) can be achieved by using an alternative definition of the matrix $\epsilon_{i,j}(p,q)$, namely $\epsilon_{i,j}(p,q) = |\ln d_i(p) - \ln d_j(q)|$. In this case, dynamic programming techniques can be used and the computational complexity is $O(N_1(N_2 - N_1))$.

Similarly, in the case of the optimization problem introduced by (12), the points $p \in S_1$ are considered as source nodes and the points $q \in S_2$ are considered as target nodes of an associated bipartite graph. Again, the optimal solution can be computed as the cost of the minimum weight assignment in the associated bipartite graph.

D. Orientation and Scale Invariance

An important property of shape descriptors is scale and orientation invariance. The very notion of shape concerns those spatial properties of an object that do not change when the object appears at different sizes or orientations.

Unless orientation is involved in the definition and extraction of features, a set of distance sets does not depend in any way on the orientation of an object in the 2-D plane: only distances matter. A set of distance sets associated with a given object can be made scale invariant by, for instance, dividing all distances in the set by the distance between the two feature points that are furthest apart, the diameter of the feature point set. As all distances change in proportion with the size, the normalized distances remain constant.

This approach to achieving scale invariance is of practical use only when the object under consideration is segmented from its background. Such a situation is assumed in the shape comparison application given in Section VI-D where scale invariance is obtained by resizing all objects to the same bounding box. In other situations, such as those illustrated in Section V-E, an object is not segmented from its background. In contrast, the very purpose of using a shape descriptor in such a situation is to test whether a given object is present in a complex scene and to separate it from the background. Under such circumstances and without any prior knowledge about the appropriate scale to be used, one can take a multiscale approach: features and distance sets are computed independently at multiple resolutions and the distance sets computed at each scale are compared with the reference sets. The plausibility of the multiscale approach has been argumented for both in biological [50] and computer [51] vision.

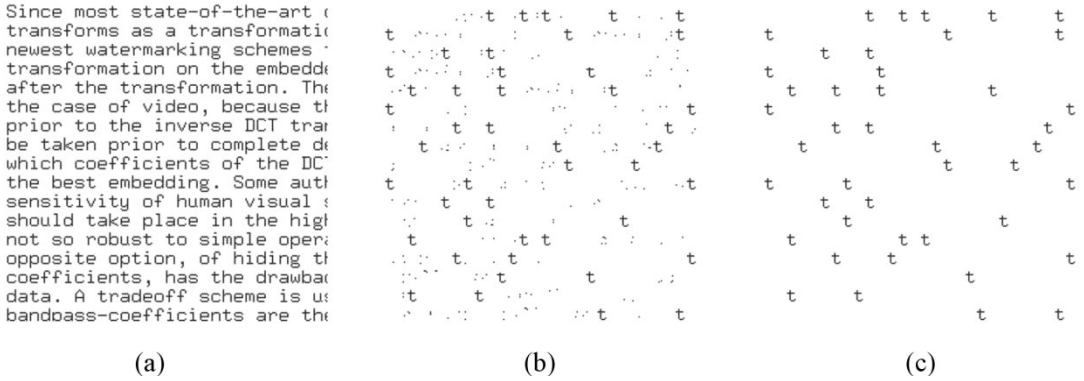

Fig. 6. (a) The set S_2 is defined by the black points in the shown piece of text. (b) The subset of points $S_2' = \{q \in S_2 \mid \exists p \in S_1, \mathcal{D}_{S_1,17;S_2,30}(p,q) = 0\}$ that remain after one application of a shape filter associated with the letter "t." (c) The subset of points $S_2'' = \{q \in S_2' \mid \exists p \in S_1, \mathcal{D}_{S_1,17;S_2',30}(p,q) = 0\}$ which remain after two applications of the filter.

V. Distance Set Shape Filters

A. Definition

Let S_1 be a set of feature points extracted from an image of a reference object. We assume that the reference object is separated from the background and there is no noise or occlusion by other objects. Let S_2 be a set of feature points extracted from another image, to be called the test image. Typically, the test image will represent a complex scene and the number of feature points obtained from this scene will be greater that the number of feature points extracted from the reference object, $|S_2| > |S_1|$. If the test image contains the reference object (at the same size and from the same view), the set S_2 will have a subset $\hat{S}_2$ which is isomorphic to S_1, in that there will be a one-to-one mapping between the sets S_1 and $\hat{S}_2$ which preserves the distances between the points within the sets.

Let $p \in S_1$ and $q \in \hat{S}_2$ be two counterpart points. The distance set $\mathcal{DS}_{S_1,N_1}(p)$ of p to its N_1 nearest neighbors in S_1 is identical with the distance set $\mathcal{DS}_{\hat{S}_2,N_1}(q)$ of q to its N_1 nearest neighbors in $\hat{S}_2$. This relation does not necessarily hold for the distance set $\mathcal{DS}_{S_2,N_1}(q)$ to the N_1 nearest neighbors of q in the bigger set S_2. If, however, q is a counterpart of p, there will be an integer number N_2, $N_2 \geq N_1$, such that $\mathcal{DS}_{S_1,N_1}(p) \subseteq \mathcal{DS}_{S_2,N_2}(q)$, where $\mathcal{DS}_{S_2,N_2}(q)$ is the distance set of q to its N_2 nearest neighbors in S_2. The latter relation can be verified using the dissimilarity $\mathcal{D}_{S_1,N_1;S_2,N_2}(p,q)$ between the distance sets $\mathcal{DS}_{S_1,N_1}(p)$ and $\mathcal{DS}_{S_2,N_2}(q)$, which must be zero according to (4). We can now use this property to determine the points of $\hat{S}_2 \subset S_1$ which are counterparts of the points of S_1. Let $\mathbf{S}_2$ be the set of all possible subsets of S_2. We define the operator $\mathcal{F}_{S_1;N_1,N_2;\theta} : \mathbf{S}_2 \mapsto \mathbf{S}_2$ associated with the set S_1 and having as parameters the number of neighbors taken into account N_1, $N_2 \in \mathbb{N}$ and a threshold value $\theta \in \mathbb{R}$ as follows:

$$\mathcal{F}_{S_1;N_1,N_2;\theta}(S_2) = \{q \in S_2 \mid \exists p \in S_1, \mathcal{D}_{S_1,N_1;S_2,N_2}(p,q) \leq \theta\}. \quad (17)$$

The subset $\hat{S}_2$ mentioned above can be determined by recursive application of the operator $\mathcal{F}$ for $\theta = 0$

$$\hat{S}_2 \subseteq \mathcal{F}^i_{S_1;N_1,N_2;0}(S_2) \quad (18)$$

where $i \in \mathbb{N}$ is the number of times $\mathcal{F}$ has to be recursively applied, $\mathcal{F}^i(S_2) = \mathcal{F}(\mathcal{F}^{i-1}(S_2))$ until $\mathcal{F}^{i+1}(S_2) = \mathcal{F}^i(S_2)$ holds. In general, i decreases with an increasing neighborhood N_1. Note that $\hat{S}_2$ is only a subset of $\mathcal{F}^i_{S_1;N_1,N_2;0}(S_2)$ because $\mathcal{F}^i_{S_1;N_1,N_2;0}(S_2)$ may contain multiple subsets which are isomorphic with S_1.

If sets of labeled distance sets are used with $\mathcal{L}$ being the set of possible labels, the operator $\mathcal{LF}$ is defined as follows:

$$\mathcal{LF}_{S_1;N_1^{(l_1)}\ldots N_1^{(l_k)};N_2^{(l_1)}\ldots N_2^{(l_k)};\theta}(S_2) = \{q \in S_2 \mid \exists p \in S_1, \mathcal{LD}_{S_1^{(l_1)},N_1^{(l_1)}\ldots S_1^{(l_k)},N_1^{(l_k)};S_2^{(l_1)},N_2^{(l_1)}\ldots S_2^{(l_k)},N_2^{(l_k)}}(p,q) \leq \theta\}. \quad (19)$$

Similar to a band-pass filter, which will retain only signal components within a certain frequency band, the proposed filter will allow to pass only those points of S_2 which, regarding their distance sets, are similar to points of S_1, and will filter out all other points. In analogy to a band-pass filter, we will call this filter the *distance set shape filter*.

B. Example of Printed Character Recognition

The following optical character recognition (OCR) example demonstrates how the set of (labeled) distance sets of a printed character can be used for finding the occurrences of that particular character in a piece of printed text.

Let S_1 be the set of points that define the printed character "t," and S_2 be the set of points of the printed text appearing in Fig. 6(a). Let $\mathcal{SDS}_{N_1}(S_1)$ be the set of N_1-distance sets of the character "t." As to the points of S_2, note that the points in the neighborhood of a given point can belong to a neighboring character. This problem is tackled by taking a bigger neighborhood around the points from S_2, $N_2 > N_1$, in determining a set of distance sets $\mathcal{SDS}_{N_2}(S_2)$. We now determine a subset $S_2' \subseteq S_2$ as follows [Fig. 6(b)]:

$$S_2' = \mathcal{F}_{S_1;N_1,N_2;0}(S_2) \quad (20)$$

or, equivalently:

$$S_2' = \{q \in S_2 \mid \exists p \in S_1, \mathcal{D}_{S_1,N_1;S_2,N_2}(p,q) = 0\}. \quad (21)$$

The set S_2' obtained in this way consists mostly of points that belong to different instances of the character "t" but there are a relatively small number of other points as well, Fig. 6(b).

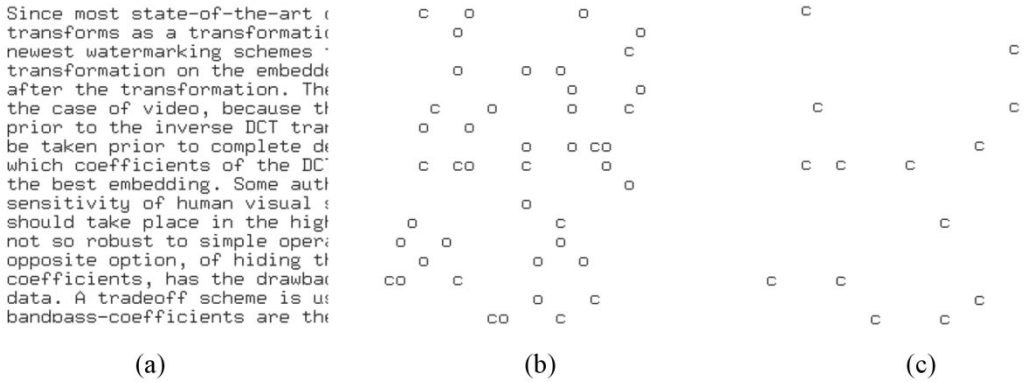

Fig. 7. (a) The set S_2 is defined by the black points in the piece of text (shown on the left). (b) The subset of points $S_2'' \subseteq S_2$ obtained by applying twice a c-shape filter that makes use of unlabeled distance sets, $S_2'' = \{q \in S_2' \mid \exists p \in S_1, \mathcal{D}_{S_1,17;S_2',30}(p,q) = 0\}$ with $S_2' = \{q \in S_2 \mid \exists p \in S_1, \mathcal{D}_{S_1,17;S_2,30}(p,q) = 0\}$ and S_1 being the set of points defining the printed character "c;" note that the instances of the character "o" are not filtered out. (c) The subset of points $Q_2' = \{q \in S_2 \mid \exists p \in S_1, \mathcal{LD}_{S_1^{(l_1)},15;S_1^{(l_2)},2;S_2^{(l_1)},27;S_2^{(l_2)},3}(p,q) = 0\}$ obtained by only one filtering iteration when labeled distance sets are used; only instances of the character "c" pass the filter.

Taking S_2' as input an applying the procedure once again a new subset of points $S_2'' \subseteq S_2' \subseteq S_2$

$$S_2'' = \mathcal{F}_{S_1;N_1,N_2;0}(S_2') = \mathcal{F}^2_{S_1;N_1,N_2;0}(S_2) \tag{22}$$

is obtained, Fig. 6(c). Only points from the instances of the character "t" that occur in the considered piece of text belong to this new subset, all the other points are filtered out.

The set of (unlabeled) distance sets $\mathcal{SDS}_N(S)$ does not offer enough discriminative power to correctly find occurrences of certain printed characters. For example, an attempt to separate the occurrences of the character "c" from the set of points S_2 of the same text (Fig. 7(a)) by applying the filtering procedure described by (20) and (22) ($N_1 = 17$, $N_2 = 30$) leads to the result presented in Fig. 7(b). Due to the fact that points defining a "c" form a subset of the set of points defining an "o," any point p of "c" will find a counterpart point q in an "o" such that $\mathcal{D}_{N_1,N_2}(p,q) = 0$. This ambiguity can be avoided by using the set of labeled distance sets $\mathcal{SDS}_{S_1^{(l_1)},N^{(l_1)};S_1^{(l_2)},N^{(l_2)}}$ of the character "c," with the labels shown in Fig. 4(a). Assigning equal weights to the two labels and applying the same filtering procedure

$$Q_2' = \{q \in S_2 \mid \exists p \in S_1, \mathcal{LD}_{S_1^{(l_1)},N_1^{(l_1)};S_1^{(l_2)},N_1^{(l_2)};S_2^{(l_1)},N_2^{(l_1)};S_2^{(l_2)},N_2^{(l_2)}}(p,q) = 0\} \tag{23}$$

leads in one iteration to the correct result, Fig. 7(c).

C. Relation to Mathematical Morphology

A shape filter based on the distance set of a single point is, in a way, related to certain operations from mathematical morphology [52]. The term "morphology" is used in many disciplines (image processing, linguistics, biology), being associated with the study of form (from Greek: "morphy"- form, figure, shape, "logos"- speech, sermon, word, reason, account). In image processing, morphological filtering operations are well known for their properties of detecting certain patterns of pixel configurations; however, such operators are mainly used for pixel-based analysis of images. A distance set shape filter can be implemented as a combination of morphological filters. Consider, for instance, a 1-distance set shape filter based on the L_1-norm and the distance set $\mathcal{DS}_{S,1}(p) = \{2\}$. This means that a feature point q will pass the filter if another feature point q' at a L_1-distance of 2 is found in the neighborhood. Fig. 8 shows a few structuring elements which, when used in an erosion operation, will detect only points which have neighbors at a L_1-distance of 2 in the given orientations. The disjunction of the results of all such erosion operations is equivalent to applying the above mentioned 1-distance set shape filter.

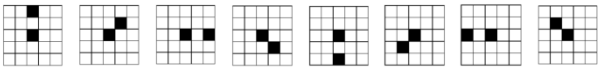

Fig. 8. Example of structuring elements that can be used with erosion operations to obtain the same result as the one obtained after applying a distance set shape filter characterized by the distance set $\mathcal{DS}_{S,1}(p) = \{2\}$.

An N-distance set shape filter can be realized as the conjunction operation of N 1-distance set shape filters which, as mentioned above, can be realized as combinations of elementary erosion operations. At a next level, a shape filter based on a set of N-distance sets can be realized as the disjunction of a collection of N-distance set operators.

Although it is possible to implement a distance set shape filter as a combination of morphological operators, the mathematical formulation in terms of distance sets is more intuitive and straightforward. Furthermore, the mathematical morphology implementation can cover only the case of a strict filter condition (20), which is a special case that we do not actually use in practice. A labeled distance set filter cannot be implemented as a combination of elementary morphological operations. For these reasons, we find the introduction of a separate term, *distance set shape filter*, appropriate and justified.

D. Application to Continuous Handwritten Text

The (labeled) distance sets and the shape filter operator presented in Sections V-A–C can be used for off-line handwritten character recognition. As an illustration, we consider a handwritten version of same piece of text presented in the printed character recognition example. In this case, let us denote by S_1 the set of points that originate from the skeleton of the handwritten character "t" and by S_2 the points originating from

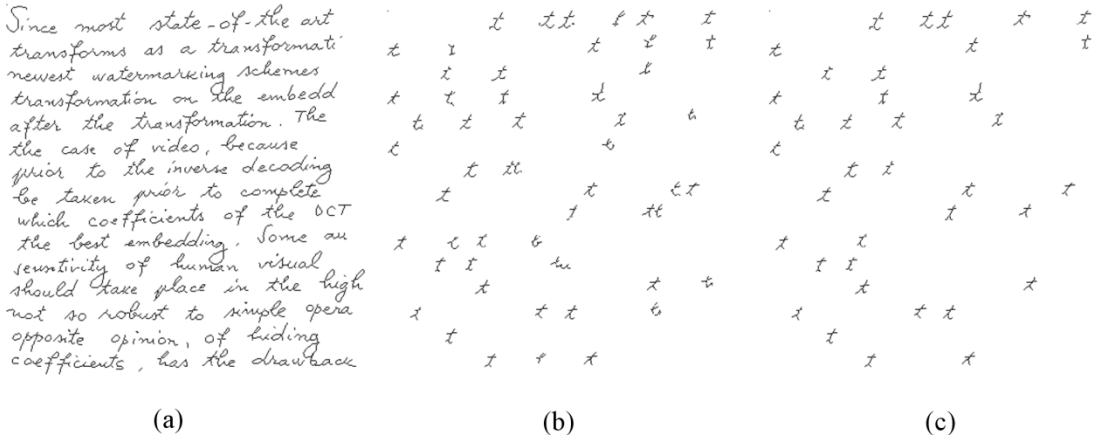

Fig. 9. (a) The set S_2 of points originating from the skeleton of a piece of handwritten text. (b) The subset of points $S_2^{(3)} = \mathcal{F}^3_{S_1;\,30,45;\,0.25}(S_2)$ remaining after applying 3 successive filtering steps with a shape filter associated with the distance set S_1 of a handwritten character t. (c) The subset of points $S_2^{(2)} = \mathcal{LF}^2_{S_1;\,28,2\,;\,43,2;\,0.25}(S_2)$ remaining after applying only 2 filtering steps with a filter associated with the labeled distance set $\mathcal{LS}_1$ of a handwritten character *t*; two types of labels, l_1—*contour point* and l_2—*junction point* were used in this case.

the skeletonized version of the handwritten text presented in Fig. 9(a). We apply iteratively the filtering procedure described by (19) and determine the subset

$$S_2^{(i)} = \mathcal{F}^i_{S_1;N_1,N_2;\theta}(S_2). \tag{24}$$

After $i = 3$ iterations, the result $S_2^{(i)}$ is no longer modified by subsequent filtering steps, Fig. 9(b); the parameters used in this example are $N_1 = 30$, $N_2 = 45$, $\theta = 0.25$. All the occurrences of character "t" are correctly found, but some other fragments of text remain and cannot be filtered out. Some of these fragments originate from the skeletons of letters "l," "b," and "h," which, for this particular handwriting style, have similar parts of the contour as the letter "t."

In order to resolve this ambiguity and improve the discrimination, additional labels have to be introduced and the set of labeled distance sets has to be used in the filtering procedure, (19). We consider the set of labels $\mathcal{L} = \{l_1, l_2\}$, with the two labels being l_1—*a contour point* and l_2—*a junction point*, and, using $N_1^{(l_1)} = 26$, $N_1^{(l_2)} = 4$, $N_2^{(l_1)} = 39$, $N_2^{(l_2)} = 6$, and $\theta = 0.25$, we determine the subset

$$S_2^{(i)} = \mathcal{LF}^i_{S_1;N_1^{(l_1)},N_1^{(l_2)};N_2^{(l_1)},N_2^{(l_2)};\theta}(S_2). \tag{25}$$

We consider the junction points as being perceptually more important than the contour points and, consequently, weight the dissimilarities differently, $w_{l_1} = 1$, $w_{l_2} = 20$ (8). After $i = 2$ filtering steps, only the occurrences of the character "t" are correctly detected, Fig. 9(c).

Since in most cases the same person can write the same character in different ways and the input text may contain occurrences of the same character written differently, a collection of (pre-stored) prototype characters representing the same letter can be used in detecting the occurrences of a particular character. The results of all filtering operations performed with different prototype characters can be "OR"-ed to produce the final result. The use of additional features can further improve the discrimination; some of these additional features, already used in different ways in handwritten text recognition, may include centers of closed contours, end-of-lines, branch points, crossing points, high curvature points, etc. [12], [53]–[55].

E. Finding Objects in Complex Scenes

One of the most challenging tasks in computer vision is the detection of a particular object in a complex scene. A widely accepted hypothesis of human object perception is that recognition and categorization performance is viewpoint dependent [56], [57], to suggest that some kind of view specific representation of objects or, at least, some component features derived from the two-dimensional views of the objects are used in recognition. Based on this two-dimensional viewpoint recognition assumption, we address in the following the problem of detecting instances of a given reference object (in this case a traffic sign) in a complex scene. The viewpoint from which we look at both the reference object and the object embedded in the complex scene is approximately the same and favors their recognition.

We created a traffic sign image database of 48 traffic scenes, containing three different traffic signs[3]. Edges from the scenes were extracted from the zero-crossings of a multiscale oriented B-spline wavelet representation [58], [59]. For edge extraction we used only the second scale of the wavelet decomposition since it represented most accurately the traffic sign edges. The wavelet chosen in our experiments decomposed the image in four different orientations for each subband. In order to eliminate small amplitude edges, a threshold condition was imposed on the gradient of the oriented edge maps (gradient be at least 15% of the maximum gradient value of the corresponding edge map). Fig. 10(a) shows a traffic image containing one of the traffic signs. Fig. 10(b) presents the combined edge map obtained after superimposing the four oriented edge maps. We assigned a different label, $\mathcal{L} = \{l_1, l_2, l_3, l_4\}$, to each of the four orientations, and a labeled distance set $\mathcal{LDS}_{S^{(l_1)},N^{(l_1)};\ldots;S^{(l_4)},N^{(l_4)}}$ was built for each point of the combined edge map by computing the distances to the first $N^{(l_k)}$ nearest neighbor points occurring in each oriented edge map, $k = 1 \ldots 4$.

We denote in the following by S_1 and S_2 the set of edge points from the combined edge map of a traffic sign template and the

[3]The image database of traffic signs and complex scenes, as well as the experimental results are available at: http://www.cs.rug.nl/~imaging

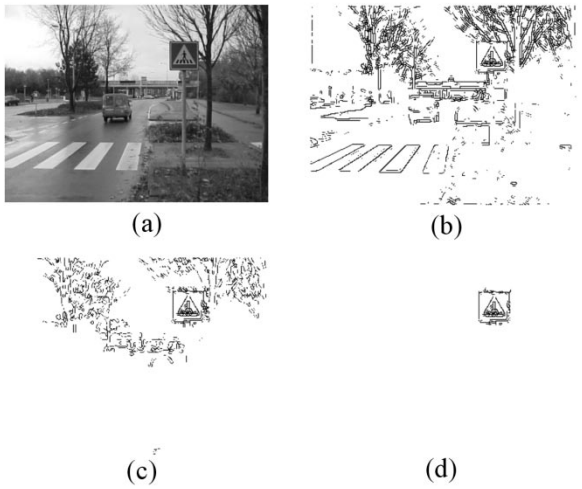

Fig. 10. (a) Traffic scene, and (b) its corresponding edge map. Labeled distance set shape filtering results after (c) one, $i = 1$, and (d) two, $i = 2$, steps.

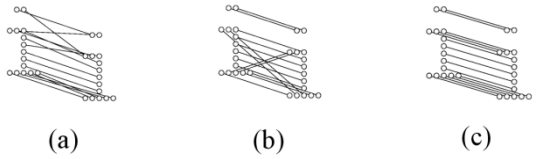

Fig. 11. Possible correspondences between the points of S_1 and S_2 of two identical copies of the printed character "i;" two nonoptimal mappings with costs (a) 0.091 and (b) 0.035 and (c) the optimal mapping with a cost 0 (the size of the neighborhood used is $N = 12$). The dissimilarity $\mathcal{E}_N(S_1, S_2)$ of the sets of distance sets associated with S_1 and S_2 is by definition equal to the cost of this optimal mapping.

complex scene, respectively. Fig. 10(c) and (d) display the results when applying iteratively the distance set filtering procedure:

$$S_2^{(i)} = \mathcal{LF}^i{}_{S_1;N_1^{(l_1)},\ldots,N_1^{(l_4)};N_2^{(l_1)},\ldots,N_2^{(l_4)};\theta}(S_2) \tag{26}$$

for $N_1^{(l_k)} = 20$, $N_2^{(l_k)} = 30$, $k = 1 \ldots 4$, and $\theta = 0.2$. For $i > 2$, the result is no longer modified by subsequent filtering steps. For all the 48 traffic scenes currently available in the database the occurrence of a given traffic signs was correctly detected.

VI. Shape Comparison

A. Method of Comparison

Assuming that the sets of distance sets $\mathcal{SDS}_N(S_1)$ and $\mathcal{SDS}_N(S_2)$ are characteristic of the geometric configurations of two sets of feature points S_1 and S_2, the condition $\mathcal{E}_N(S_1, S_2) = 0$ can be used to check whether these two sets of feature points are identical regarding their geometric configurations (up to any transformation that preserves a set of distance sets). In the case that the quantity $\mathcal{E}_N(S_1, S_2)$ is not equal to zero, it can still be used as an indicator of how (dis)similar two sets of feature points are.

For illustration, we consider the sets S_1 and S_2 defined by the points of two identical copies of the printed character "i." Fig. 11 shows three possible mappings of S_1 onto S_2. A cost can be assigned to each such mapping according to (11) or (14). Fig. 11(c) shows the optimal mapping with zero cost found after solving the optimization problem defined by (12) or (15), respectively.

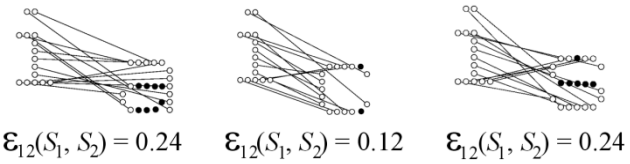

Fig. 12. Optimal mappings of a set S_1 onto different sets S_2 and the dissimilarities between the corresponding sets of distance sets. The points of S_2 that do not have counterparts in S_1 are shed black; each such point increases the cost of the optimal mapping by a constant C.

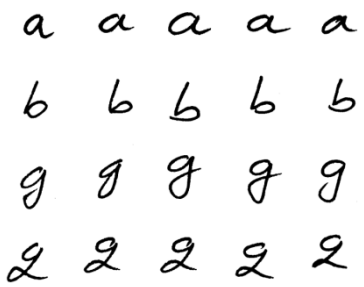

Fig. 13. Examples of handwritten characters used for shape comparison.

Fig. 12 shows the optimal mappings between points of different printed characters and the dissimilarities between the corresponding associated sets of distance sets. For computing the cost of mappings, the value of the constant C in (11) was chosen as the average pair-wise dissimilarity of two points:

$$C = \frac{1}{|S_1||S_2|} \sum_{q \in S_2} \sum_{p \in S_1} \mathcal{D}_{S_1,N;S_2,N}(p, q). \tag{27}$$

B. Application to Handwritten Character Recognition

In a more elaborate example, a database of 286 characters, 11 characters for each of the 26 letters, handwritten by one person was used, Fig. 13. The skeleton for each character was extracted and the points from the skeleton were further considered as feature points in the shape comparison procedure.

For each of the 26 categories, one of the 11 instances was randomly selected and considered as a test character. The remaining 260 characters were considered as references. The dissimilarity between each of the test characters and all reference characters was computed according to (11); again, the constant C was calculated as in (27).

The dissimilarity between any character and its nearest neighbor from the same class is considerably smaller than the dissimilarity of this character and its nearest neighbor from a different class. Fig. 14 presents a histogram of the minimum dissimilarity values. The values obtained for characters from the same class cluster in the black bins of the histogram, while the values obtained for characters from different classes fill the white bins. The bimodal nature of the histogram allows classification by thresholding.

C. Application to Object Recognition Based on 2-D Views

In the following, we evaluate the shape comparison procedure for recognition of objects based on their 2-D appearances. We used the COIL-20 database [60]. Each object from this database

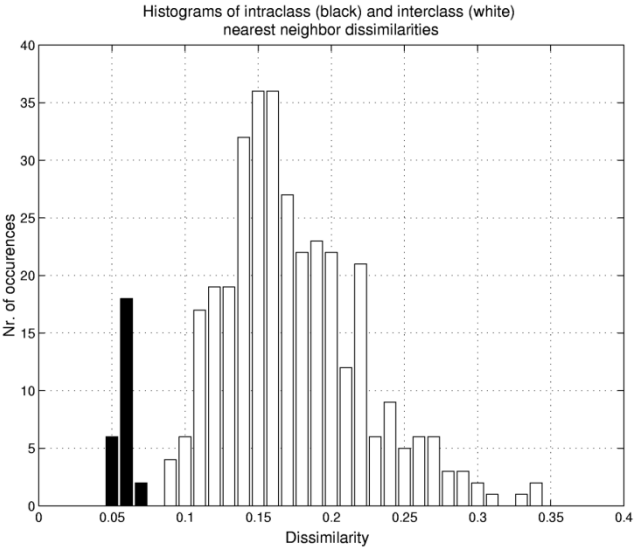

Fig. 14. Histograms of the minimum values of dissimilarities between handwritten test characters and reference characters from the same (black) and different (white) classes.

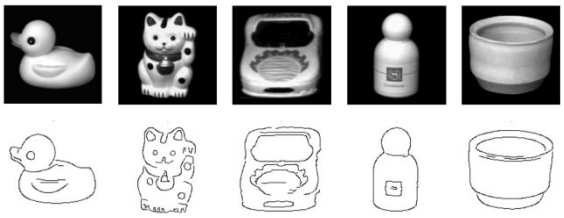

Fig. 15. Examples of contour maps extracted from COIL-20 images.

is photographed from 72 different views around the object, corresponding to equally spaced 5° angular offsets. The distance between the object and the camera is approximately the same.

We start by extracting contours using a Gabor energy operator augmented with isotropic inhibition [11], Fig. 15.

A distance set is constructed for each of the detected contour points , and the dissimilarity between two contour maps defined by two such sets of points is computed according to (11). The cost C was calculated as in (27). The number of neighbors taken into account within a distance set is $N = 0.3 \min(|S_1|, |S_2|)$, where $|S_1|$ and $|S_2|$ are the numbers of feature points in two contour maps that are compared. The dissimilarities of all contour maps to a subset of 360 contour maps (prototypes) corresponding to 18 views per object, equally spaced at angular offsets of 20°, were analyzed. For all contour maps, the nearest neighbor was always a prototype contour map from the same class (0% misclassification).

The number of prototypes can be reduced because some of the objects have approximately the same appearance from all viewpoints. For such an object, e.g., the cup shown in the last column of Fig. 15, one can take a lesser number of prototypes. More generally, one can consider prototype selection as an optimization problem in which one tries to jointly minimize the number of prototypes and the misclassification error. For instance, Belongie *et al.* [61] used a modified k—*means* clustering algorithm for adaptively selecting prototypes. They report 2.4% misclassification by using only 80 prototypes. More intricate clustering algorithms may lead to an even smaller number of prototypes while achieving a reduced classification error. When such an algorithm is employed, it is, however, difficult to assess the quality of the shape comparison method: a reduced classification error may be more a result of smart clustering than due to the discrimination properties of the shape comparison method. We consider clustering methods for optimal prototype selection rather a study in itself that is beyond the scope of this paper.

Fig. 16. Examples from the MPEG-7 shape database for four different categories.

D. Application to MPEG-7 Shape Database Retrieval

In the following experiment we measure the similarity-based retrieval performance for a database of 1400 objects used in the MPEG-7 Core Experiment CE-Shape-1 part B [62]. This database consists of 70 shape categories, with 20 objects per category, Fig. 16. Each object is used successively as a query object and compared with all 1400 objects; the retrieval rate for the query object is measured by counting the number of objects from the same category which are found in the first 40 most similar matches (bulls-eye test). The maximum number of objects from the same category which can appear in the first 40 retrieved objects is 20. The total number of possible correct matches when all 1400 objects are used in turn as queries is thus 28.000. The overall retrieval rate is computed as the ratio of the total number of actual correct matches and the total number of possible correct matches.

Since the original database contains images of objects of different sizes, we first rescaled every object to the same minimum of width and height of its bounding box. The contours were extracted with an edge detector operator and the contour points were further considered as feature points of type l_1—*contour*; 150 to 300 such points were extracted per object. The center of the bounding box of the object was considered as an additional feature point, of type l_2—*center*, and the associated labeled distance sets, $\mathcal{LDS}_{S^{(l_1)},N^{(l_1)};S^{(l_2)},N^{(l_2)}}(p)$, were computed only for the *contour* points. Let S_1 and S_2 denote the sets of points representing a query object and a test object, respectively. The dissimilarity between labeled distance sets for two points $p \in S_1$ and $q \in S_2$ was computed using $N^{(l_1)} = 0.3 \min(|S_1|, |S_2|)$ nearest neighbors of type contour point and $N^{(l_2)} = 1$ center point. Equal weights were used for the distance sets to the contour points and the distance to the center point in computing the dissimilarity $\mathcal{LD}_{S_1^{(l_1)},N^{(l_1)},S_1^{(l_2)},N^{(l_2)};S_2^{(l_1)},N^{(l_1)},S_2^{(l_2)},N^{(l_2)}}(p,q)$ according to (8).

The dissimilarity between the sets of labeled distance sets was determined according to (15). Correspondences $p \mapsto M(p)$ for which the dissimilarity exceeded a certain threshold value,

$$\mathcal{LD}_{S_1^{(l_1)},N^{(l_1)},S_1^{(l_2)},N^{(l_2)};S_2^{(l_1)},N^{(l_1)},S_2^{(l_2)},N^{(l_2)}}(p, M(p)) > 0.3,$$

originate from configurations of points which are perceptually very different. We considered these points as completely dissimilar, and set the value of the dissimilarity to

$$\mathcal{LD}_{S_1^{(l_1)},N^{(l_1)},S_1^{(l_2)},N^{(l_2)};S_2^{(l_1)},N^{(l_1)},S_2^{(l_2)},N^{(l_2)}}(p, M(p)) = 1.$$

The above operation can be viewed as an additional thresholding operation performed on the values of dissimilarity. For situations in which $|S_1| > |S_2|$, the sets of points were first swapped and then the comparison procedure was applied. The penalty cost C for those points of S_2 which are not images of points of S_1 was set to a constant value, $C = 1$.

The time needed for computing the dissimilarity between two objects, each being described by approximately 250 points and having associated distance sets of 100 distances, is about 0.7 s on a regular Pentium III/667 MHz computer. With this experimental setup, we achieved an overall retrieval rate of 78.38%. Previous studies report 78.17% [21], 76.51% [20], 76.45% [62] and 75.44% [19]. One possible reason for achieving improved results is that distance sets are invariant with respect to reflections, whereas shape contexts are not. Examples of similar shapes where this can be a problem are shown in the first row of Fig. 16.

VII. Summary and Conclusions

We proposed a new local image point descriptor, namely the (labeled) distance set, which is determined by the spatial configuration of feature points in the surrounding of a given point. We formulated an appropriate dissimilarity measure between two (labeled) distance sets and, based on it, we defined a novel shape filter. Relying upon the geometric arrangement of features originating from a reference object, the (labeled) distance sets shape filter is able to segment instances of the reference object in images even when they are embedded in a complex environment, have a deformed appearance or are partially occluded by other image structures.

Furthermore, by introducing an appropriate dissimilarity measure between two sets of (labeled) distance sets, we proposed a new shape comparison method. Formulated as a minimum cost assignment in an associated bipartite graph, the proposed shape comparison method delivers a reliable shape dissimilarity measure which can be used for shape classification.

We demonstrated the applicability of the distance set shape filter for segmentation, and illustrated its effectiveness for letter segmentation in handwritten character recognition and for the isolation of traffic signs in complex traffic scenes. We evaluated the performance of the distance set shape comparison method in the context of three other applications: handwritten character recognition, object recognition and image database retrieval. For the latter, a retrieval experiment was performed on a MPEG-7 shape database. Our method outperforms all shape comparison methods previously reported in the literature.

The proposed shape comparison method is based exclusively on a dissimilarity measure computed from point correspondences. This method does not explicitly model transformations between objects, is robust to local shape deformations and shows reduced computational complexity.

Acknowledgment

The authors thank L. J. Latecki for providing them with the MPEG-7 silhouette database.

References

[1] S. Zeki, *A Vision of the Brain*. Cambridge, MA: Blackwell Scientific, 1993, pp. 312–320.
[2] D. H. Hubel and T. N. Wiesel, "Receptive fields. binocular interaction, and functional architecture in the cat's visual cortex," *J. Phys. (Lond.)*, vol. 160, pp. 106–154, 1962.
[3] ——, "Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat," *J. Phys. (Lond.)*, vol. 28, pp. 229–289, 1965.
[4] ——, "Receptive fields and functional architecture of monkey striate cortex," *J. Phys. (Lond.)*, vol. 195, pp. 215–243, 1968.
[5] ——, "Sequence regularity and geometry of orientation columns in the monkey striate cortex," *J. Comput. Neurol.*, vol. 158, pp. 267–293, 1974.
[6] D. H. Hubel, "Explorations of the priurary visual cortex, 1955–1978," *Nature*, vol. 299, pp. 515–524, 1982.
[7] A. Dobbins, S. W. Zucker, and M. S. Cynader, "Endstopped neurons in the visual cortex as a substrate for calculating curvature," *Nature*, vol. 329, pp. 438–441, 1987.
[8] E. Peterhaus and R. von der Heydt, "The role of end-stopped receptive fields in contour perception," in *New Frontiers in Brain Research, Proc. 15th Goettingen Neurobiology Conf.*, N. Elsner and O. Creutzfeld, Eds., 1987, p. 29.
[9] M. Versavel, G. A. Orban, and L. Lagae, "Responses of visual cortical neurons to curved stimuli and chevrons," *Vis. Res.*, vol. 30, pp. 235–248, 1990.
[10] H. von der Heydt, F. Peterhans, and M. R. Dürsteler, "Periodic-pattern-selective cells in monkey visual cortex," *J. Neurosci.*, vol. 12, pp. 1416–1434, 1992.
[11] C. Grigorescu, N. Petkov, and M. Westenberg, "Contour detection based on nonclassical receptive field inhibition," *IEEE Trans. Image Processing*, to be published.
[12] O. D. Trier, A. K. Jain, and T. Taxt, "Feature extraction methods for character recognition—A survey," *Pattern Recognit.*, vol. 29, no. 4, pp. 641–662, 1996.
[13] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer, "A robust visual method for assessing the relative performance of edge-detection algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 1338–1359, Dec. 1997.
[14] T. Randen and J. H. Husøy, "Filtering for texture classification: A comparative study," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 291–310, Apr. 1999.
[15] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. Pacific Grove, CA: Brooks/Cole, 1999.
[16] S. E. Grigorescu, N. Petkov, and P. Kruizinga, "Comparison of texture features based on Gabor filters," *IEEE Trans. Image Processing.*, vol. 11, pp. 1160–1167, Oct. 2002.
[17] S. Loncaric, "A survey of shape analysis techniques," *Pattern Recognit.*, vol. 31, no. 8, pp. 983–1001, 1998.
[18] L. J. Latecki and R. Lakamper, "Shape similarity measure based on correspondence of visual parts," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 1185–1190, Oct. 2000.
[19] F. Mokhtarian, S. Abbasi, and J. Kittler, "Efficient and robust retrieval by shape content through curvature scale space," in *Image Databases and Multi-Media Search*, A. W. M. Smeulders and R. Jain, Eds, Singapore: World Scientific, 1997, pp. 51–58.
[20] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 4, pp. 509–522, 2002.
[21] T. B. Sebastian, P. N. Klein, and B. B. Kimia, "On aligning curves," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, pp. 116–125, Jan. 2003.

[22] D. Huttenlocher, G. Klanderman, and W. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 850–863, Sept. 1993.
[23] A. Cross and E. Hancock, "Graph matching with a dual-step EM algorithm," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 11, pp. 1236–1253, 1998.
[24] P. Kruizinga and N. Petkov, "Optical flow applied to person identification," in *Proc. EUROSIM Conf. Massively Parallel Processing Applications and Development*, 1994, pp. 871–878.
[25] L. Wiskott, J. M. Fellous, N. Kruger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 7, pp. 775–779, 1997.
[26] Y. Amit, D. Geman, and K. Wilder, "Joint induction of shape features and tree classifiers," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 11, pp. 1300–1305, 1997.
[27] D. Huttenlochcr, H. Lilian, and C. Olson, "View-based recognition using an eigenspace approximation to the Hansdorff measure," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 951–955, Sept. 1999.
[28] S. Belongie, J. Malik, and J. Puzicha, "Matching shapes," in *Proc. 8th IEEE Int. Conf. Computer Vision*, vol. 1, Vancouver, BC, Canada, 2001, pp. 454–461.
[29] Y. Amit and D. Geman, "Computational model for visual selection," *Neural Comput.*, vol. 11, no. 7, pp. 1691–1715, 1999.
[30] R. C. Veltkamp and M. Hagedoorn, "State of the Art in Shape Matching," Utrecht Univ., Tech. Rep. UU-CS-1999-27, 1999.
[31] M. Fischler and R. Elschlager, "The representation and matching of pictorial structures," *IEEE Trans. Comput.*, vol. C-22, no. 1, pp. 67–92, 1973.
[32] L. S. Shapiro and J. M. Brady, "Feature-based correspondence: An eigenvector approach," *Image Vis. Comput.*, vol. 10, no. 5, pp. 283–288, 1992.
[33] S. Sclaroff and A. Pentland, "Modal matching for correspondence and recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 545–561, June 1995.
[34] H. Chui and A. Rangarajan, "A new algorithm for nonrigid point matching," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000, pp. 44–51.
[35] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2D and 3D point matching—pose estimation and correspondence," *Pattern Recognit.*, vol. 31, no. 8, pp. 1019–1031, 1998.
[36] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 4, pp. 377–388, 1996.
[37] G. Wahha, *Spline Models for Observational Data*. Philadelphia, PA: SIAM, 1990.
[38] E. G. M. Petrakis, A. Diplaros, and E. Milios, "Matching and retrieval of distorted and occluded shapes using dynamic programming," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, pp. 1501–1516, Nov. 2002.
[39] C. Olson and D. Huttenlocher, "Automatic target recognition by matching oriented edge pixels," *IEEE Trans. Image Processing*, vol. 6, pp. 103–113, Jan. 1997.
[40] Y. Amit and A. Kong, "Graphical templates for image registration," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 225–236, Mar. 1996.
[41] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Comput.*, vol. 9, pp. 1545–1588, 1997.
[42] D. M. Gavrila, "Multi-feature hierarchical template matching using distance transforms," in *IEEE Int. Conf. Pattern Recognition (JCPR '98)*, Brisbane, Australia, 1998, pp. 439–444.
[43] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*. New York: Springer, 1997.
[44] J. L. Bentley, "K-D trees for semidynamic point sets," in *Proc. 6th ACM Annu. Symp. Computational Geometry*, 1990, pp. 187–497.
[45] ——, "Decomposable searching problems," *Inform. Process. Lett.*, vol. 8, pp. 244–251, 1979.
[46] S. Nene and S. Nayar, "A simple algorithm for nearest neighbor search in high dimensions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 989–1003, Sept. 1997.
[47] K. Mehlhorn and S. Naher, *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[48] C. Papadimitriou and K. Stieglitz, *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
[49] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, pp. 325–340, 1987.
[50] J. J. Koenderink and A. J. van Doorn, "Visual detection of spatial contrast; influence of location in the visual field, target extent and illuminance level," *Biol. Cybern.*, pp. 157–167, 1978.
[51] P. J. Burt, "Smart sensing with a pyramid vision machine," *Proc. IEEE*, vol. 76, no. 8, pp. 1006–1015, 1988.
[52] J. Serra, *Image Analysis and Mathematical Morphology*. New York: Academic, 1982.
[53] J. J. Brault and R. Plamondon, "Segmenting handwritten signatures at their perceptually important points," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 953–957, Sept. 1993.
[54] R. Plamondon and S. Srihari, "On-line and off-line handwritten recognition: A comprehensive survey," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 63–84, Jan. 2000.
[55] G. Nagy, "Twenty years of document image analysis in PAMI," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 38–62, Jan. 2000.
[56] M. Tarr, P. Williams, W. Hayward, and I. Gauthier, "Three-dimensional object recognition is viewpoint dependent," *Nature Neurosci.*, vol. 1, no. 4, pp. 275–277, 1998.
[57] V. Blanz, M. Tarr, and H. Bülthoff, "What objects attributes determine canonical views," *Perception*, vol. 28, pp. 575–599, 1999.
[58] Y. P. Wang and S. L. Lee, "Scale-space derived from B-splines," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 10, pp. 1040–1055, 1998.
[59] Y. P. Wang, "Image representation using multiscale differential operators," *IEEE Trans. Image Processing*, vol. 8, pp. 1757–1771, Dec. 1999.
[60] H. Murase and S. Nayar, "Visual learning and recognition of 3-D objects from appearance," *Int. J. Comput. Vis.*, vol. 14, no. 1, pp. 5–24, 1995.
[61] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, pp. 509–522, Apr. 2002.
[62] L. J. Latecki, R. Lakamper, and U. Eckhardt, "Shape descriptors for nonrigid shapes with a single closed contour," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000, pp. 424–429.

Cosmin Grigorescu (S'02) received the Dipl.Eng. and M.S. degrees in computer science from Politehnica University of Bucharest, Romania, in 1995 and 1996. In 2003, he received the Ph.D. degree from the Department of Computing Science, University of Groningen, The Netherlands, where he is currently a postdoc.

From 1996 to 1998, he taught and performed research at the Automatic Control and Computers Faculty, Politehnica University. His areas of research are image processing, computer vision, and pattern recognition, with emphasis on models inspired from human vision.

Nicolai Petkov is Scientific Director with the Institute of Mathematics and Computing Science, University of Groningen, The Netherlands, where he is Chair of Parallel Computing and Intelligent Systems. He is author of two books and 90 scientific publications. His current research interests are in the area of computer simulations of the visual system, making links between computer vision, neurophysiology, psychophysics, and arts.